International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.208

(01/2021)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Application for Interactive Digital Television – Part 1

Harmonization of integrated broadcast-broadband digital television application control framework

Recommendation ITU-T J.208

Recommendation ITU-T J.208

Harmonization of integrated broadcast-broadband digital television application control framework

Summary

Recommendation ITU-T J.208 specifies methods for harmonization of integrated broadcast-broadband (IBB) systems or their application environment by identifying commonalities across IBB systems and maximizing portability of IBB applications.

Recommendations ITU-T J.207 and ITU-R BT.2075 provide guidance to IBB service providers on the selection of IBB systems for their use. In general, IBB systems comprise various hardware and software components, and tailoring them for a specific IBB system leads to extensive development by manufacturers. On the other hand, in order to deploy the services in a wider area, IBB service providers need to develop their IBB applications for each IBB system if operators or broadcasters on which service providers intend to deploy their services use a different IBB system to that on which the services are originally deployed.

Thus, it is beneficial for both IBB service providers and manufacturers of IBB capable reception devices such as set-top boxes to harmonize IBB systems. Note that the use of companion devices is already a part of IBB services.

In addition, information about the IBB application environment for the implementation of IBB application software for other IBB systems is useful in the deployment of the same service on different IBB systems.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.208	2021-01-13	9	11.1002/1000/14584

Keywords

DTV, IBB system, integrated broadcast-broadband, ITU-T J.207, ITU-R BT.2075, ITU-R BT.2267

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Introduction

[b-ITU-T J.200], [b-ITU-T J.201] and [b-ITU-T J.202] are Recommendations for the harmonization of interactive television (TV) systems and content. [b-ITU-T J.201] specifies the common core of presentation engines, intermediate language for the conversion of interactive content, and the runtime environment required to provide multiple language options. [b-ITU-T J.202] specifies the common core of the execution engines. [b-ITU-T J.200] specifies the high-level architecture of the common core.

As described in [b-ITU-T J.205], integrated broadcast-broadband (IBB) systems combine both broadcast and broadband. By the use of broadband, IBB systems enable more flexible services than interactive TV systems. Further, the use of broadband allows IBB systems to easily offer international services.

The IBB systems included in [ITU-T J.207], namely HbbTV [b-ETSI TS 102 796] [b-ETSI TS 102 809], Hybridcast [b-IPTVFJ STD-0010] Hybridcast [b-IPTVFJ STD-0010] [b-IPTVFJ STD-0011] [b-IPTVFJ STD-0013] [b-ARIB STD-B24] [b-ARIB STD-B62], HbbTV 2.0 [b-ETSI TS 102 796] [b-ETSI TS 102 809], TOPSmedia [b-TTAK.KO-07.0111] and Ginga [b-ABNT NBR 15606-2] [b-ABNT NBR 15606-11], TOPSmedia [b-TTAK.KO-07.0111], and Ginga-HTML5 [b-ABNT NBR 15606-2] [b-ABNT NBR 15606-11] are all systems based on hypertext markup language (HTML). However, applications in these systems are not compatible, and therefore, construction of individual application(s) for the same services in each system is required. To exchange IBB services offered by the IBB systems considered in [ITU-T J.207] and to deploy these services in an area where other systems are used, it is important to harmonize the IBB systems to ensure compatibility with IBB applications and interoperability across systems.

This Recommendation specifies methods to harmonize the IBB systems in [ITU-T J.207] based on an analysis of these systems from several perspectives.

Recommendation ITU-T J.208

Harmonization of integrated broadcast-broadband digital television application control framework

1 Scope

This Recommendation specifies methods to harmonize IBB systems or their application environment by identifying commonalities across the IBB systems in [ITU-T J.207] and [ITU-R BT.2075] and maximizing portability of IBB applications for those systems.

[ITU-T J.207] and [ITU-R BT.2075] provide guidance on the selection of integrated broadcast-broadband (IBB) systems to those who intend to deploy IBB services. In addition to the selection of an IBB system specification, it is desirable for IBB service providers to maximize commonality or portability of their IBB applications on multiple systems to expand their service availability. It is also desirable for IBB capable receivers and their companion device manufacturers to maximize commonality of IBB systems.

NOTE – The terms "IBB system" and "IBB service" used in [ITU-R BT.2075] should be read as "IBB DTV system" and "IBB DTV service", respectively, to align with [b-ITU-T J.205].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T J.207] Recommendation ITU-T J.207 (2019), *Specification for an integrated broadcast and broadband digital television application control framework.*

[ITU-R BT.2075] Recommendation ITU-R BT.2075-1 (2017), *Integrated broadcast-broadband system*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 service associated IBB application [b-ITU-T J.205]: An application that is part of the integrated broadcast and broadband (IBB) DTV service tuned by the user at a given time.

3.1.2 stand-alone IBB application [b-ITU-T J.205]: Resident or downloaded integrated broadcast and broadband (IBB) installable application that is not part of an IBB DTV service. Such an application can be created by an authorized IBB application provider, and typically delivered through the application repository.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AIT Application Information Table
Ajax Asynchronous JavaScript + extensible markup language
API Application Programming Interface
DTV Digital Television
DVB Digital Video Broadcasting
HDTV High-Definition Television
HTML Hypertext Markup Language
IBB Integrated Broadcast-Broadband
JSON JavaScript Object Notation
REST Representational State Transfer
TV Television
UI User Interface
XML Extensible Markup Language

5 Conventions

None.

6 Approaches for harmonization

6.1 Possible architecture of harmonized application execution environment

There are several methods to provide an IBB service by using the same application on multiple platforms.

1) Abstraction of execution environment

This approach provides an abstraction layer in the execution environment in a receiver. The abstraction layer unifies the underlying functionalities into a set of application programming interfaces (APIs) so that the upper-layer software can be individually implemented. Further, the multiple upper-layer software can be used for its specific purposes. Figure 1 shows how the abstraction layer works for the harmonization of IBB systems.

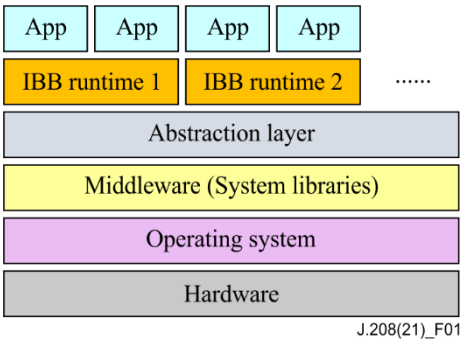

Figure 1 – Basic IBB receiver structure with an abstraction layer

In Figure 1, multiple IBB execution environments (runtime) can be implemented on the same abstraction layer. Compared with the case in [ITU-R BT.2075] and [ITU-T J.207], each IBB runtime in Figure 1 corresponds to each IBB system included in those Recommendations. Since the IBB

systems in this Recommendation are based on hypertext markup language (HTML), an HTML browser can be considered to play part of the abstraction layer role. In this case, each IBB runtime is implemented in the form of a JavaScript library. A combination of web browser and JavaScript library leads to very powerful expandability as seen in service implementations using asynchronous JavaScript plus extensible markup language (XML)(Ajax). However, the functionality of this combination is limited by that of the web browser. In the IBB digital television (DTV) receiver specified in [b-ITU-T J.205], IBB specific functionalities that are not normally installed in a web browser are achieved by middleware or by a combination of middleware and a dedicated web browser. Therefore, an abstraction layer or web browser should be designed to allow communication between IBB runtime and middleware.

2) Application conversion

This approach converts an application from one IBB system to another in order to run on other systems. In other words, this implies porting an application to other systems. Such porting can be achieved when a functional equivalent exists among the systems considered. In the IBB systems considered in [ITU-R BT.2075] and [ITU-T J.207], many equivalent functions are found. For example, the application launch process for service-associated applications in each system is very similar. For functionalities that exist only in specific IBB system(s), it is necessary to use other approaches to achieve the same functionalities on other systems. However, functionalities covered by this approach can be executed on existing IBB receivers, which is very cost effective on the receiver side.

3) Server side or cloud rendering

Considering the rapid development of broadband technologies, the installation of a virtual receiver on the server side has recently become feasible. Figure 2 shows how a virtual receiver works.

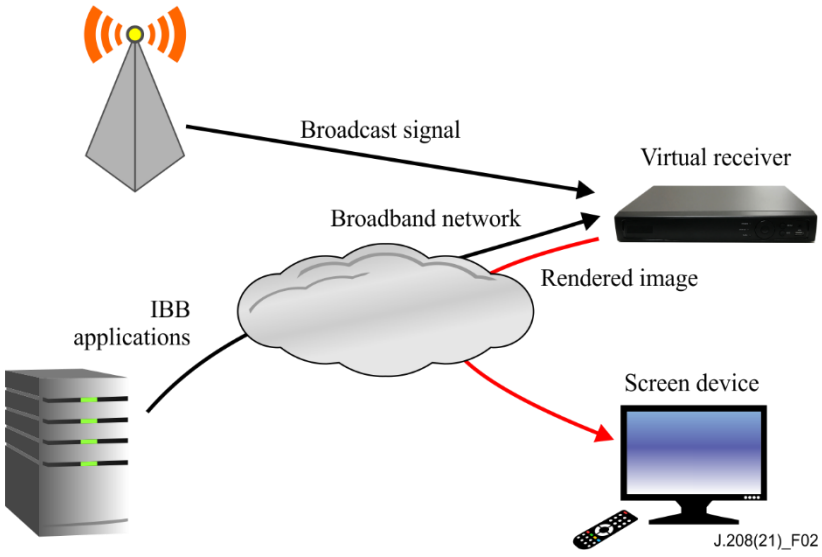

Figure 2 – Server side or cloud rendering

A virtual receiver receives a broadcast signal and executes the IBB applications. The virtual receiver is located on the server(s), which can be in the cloud, and does not have a directly connected screen. The virtual receiver decodes audio visual broadcast signals and renders graphics as instructed by the IBB applications on its video memory. The receiver transfers the rendered images in the video memory to a screen device over a broadband network. The receiver also responds to commands by a remote controller, which communicates via a device for the screen or via the broadband network.

In this approach, a virtual receiver is a receiver that is compliant with a specific IBB system. Applications for such a specific IBB system run on this virtual receiver. When applications for other IBB systems need to be run, the corresponding virtual receivers should be used.

The result of the rendering, which is an audio and a composition of a video image and graphics, is presented on a screen over the broadband network in order to minimize the role of a local receiver. Since the currently available bandwidth of a broadband network is insufficient for uncompressed high-definition television (HDTV) video images, re-compression may be required to deliver them from the virtual receiver to the screen device. Such re-compression causes a degradation of picture quality and a delayed response to user manipulation on a remote controller. Further, a special application to make an IBB receiver work as a screen device on each IBB system is required. Therefore, while this approach ensures the execution of IBB applications, the user experience strongly depends on broadband network conditions.

6.2 Commonality of application type

In [ITU-T J.205], two major types of applications are specified, namely service-associated and stand-alone. In a stand-alone application, there are various requirements for each IBB system. These requirements are mostly achieved by IBB system middleware, which leads to differences across IBB systems. On the other hand, behavioural models for service-associated applications are similar to all the IBB systems in [ITU-R BT.2075] and [ITU-T J.207].

For example, when comparing Hybridcast [b-IPTVFJ STD-0010] [b-IPTVFJ STD-0011] [b-IPTVFJ STD-0013] [b-ARIB STD-B24] [b-ARIB STD-B62], HbbTV 2.0 [b-ETSI TS 102 796] [b-ETSI TS 102 809], TOPSmedia [b-TTAK.KO-07.0111] and Ginga [b-ABNT NBR 15606-2] [b-ABNT NBR 15606-11], they support both service-associated and stand-alone IBB applications. Table 1 shows the respective application types.

Table 1 – Application types

	Service-associated IBB application	**Stand-alone IBB application**
Hybridcast	Broadcast-oriented managed application	Non-broadcast-oriented managed application
HbbTV 2.0	Broadcast-related application	Broadcast-independent application
TOPSmedia	Broadcast-activated application	Broadcast-inactivated application
Ginga	Associated local client	Stand-alone local client

There are differences in the application behaviour specified in each system for stand-alone IBB applications. For Hybridcast, it is supposed that non-broadcast-oriented managed applications are primarily launched by an application launcher installed in a receiver, while broadcast-independent applications in HbbTV 2.0 are primarily launched by either broadcast-related or other broadcast-independent applications. In HbbTV 2.0, an application launcher in a receiver is optional. For TOPSmedia, broadcast-inactivated applications can be launched by broadcast-activated applications or other broadcast-inactivated applications. TOPSmedia also supports an application launcher in the case of pre-installed applications, also known as store applications. In Ginga, stand-alone local clients may be authorized (or not) to access broadcast resources. These applications may be launched from any other application by calling the Ginga WebServices Remote APIs or from the application catalogue user interface (UI).

These systems behave similarly regarding the life cycle of stand-alone IBB applications, which is not controlled by broadcasters. A non-broadcast-oriented managed application in Hybridcast and a broadcast-independent authorized application in Ginga are allowed to access broadcast resources when a broadcaster whose offering resource is about to be accessed permits the application to do so. That is, each broadcaster can transmit access permission information to each application of these types. Broadcast-independent and broadcast-inactivated applications in HbbTV 2.0 and TOPSmedia are not allowed to access broadcast resources. However, a broadcast-independent application can

transition to a broadcast-related application. Once a broadcast-independent application transitions to a broadcast-related application, it is allowed to access broadcast resources.

In HbbTV 2.0, an application is allowed to transition between broadcast-related and broadcast-independent applications in both directions under the control of broadcasters. In Hybridcast, a broadcast-oriented managed application can transition to a non-broadcast-oriented application only. That is, a non-broadcast-oriented managed application is not allowed to transition to a broadcast-oriented managed application. It is considered that one reason for this difference between Hybridcast and HbbTV 2.0 is a difference in capability of parallel execution of multiple applications. Hybridcast is designed to execute multiple non-broadcast-oriented managed applications in addition to a broadcast-oriented managed application simultaneously on the same receiver.

Meanwhile, TOPSmedia only allows the transition from broadcast-activated to broadcast-inactivated applications under the control of broadcasters; it also supports one application execution at a given time like HbbTV 2.0.

In Ginga, applications may be allowed to transition to another type in the same lifecycle, depending on the implementation. Additionally, an application can freely launch applications of another type.

Considering the difference in the behaviour of stand-alone applications of each system, focusing on service-associated IBB applications is a reasonable approach for further analysis.

6.3 Practical approach

On the basis of the analysis described in clauses 6.1 and 6.2, it is considered that the application conversion described in clause 7 for service-associated IBB applications is the most practical approach for the harmonization of IBB systems.

7 Comparison of application environment of various IBB systems

This clause describes a comparison of the application environments of IBB systems in both aspects of syntax and behaviour. Clause 7.1 describes the comparison in terms of the syntactical aspects. Clause 7.2 compares life cycle aspects. As for application conversion, differences in behaviour among each IBB systems should be carefully considered.

7.1 Comparison of additional objects, properties and methods

As Hybridcast, HbbTV 2.0, TOPSmedia and Ginga-HTML5 are based on HTML5, functions provided by APIs, elements, attributes, etc. defined in HTML5 and other related Recommendations of the World Wide Web Consortium, e.g., encrypted media extensions, are considered common between these systems.

On the other hand, each system specifies additional objects and their properties and methods,[1] elements,[2] attributes, etc. to provide functionalities needed for IBB systems. Those additions are not common between the systems. However, considering similarity of behaviours of service-associated applications in these systems, there may be similarity or equivalency in those additions in the systems. Table 2 should be equivalent additional objects, properties and methods that have been identified by detailed analysis of the systems. Note that Table 2 does not take into account combination of additional methods to provide equivalent function among systems, and that properties and methods specific to a system in the object are not listed. For example, identification of StreamEvent is incompatible between the systems, because it is strongly tied to a multiplexing scheme, i.e., the ARIB

[1] In this context, "property" and "method" are the terms used in object-oriented programming. A property represents data or an event relevant to an object. A method is a function to give a behaviour of an object.

[2] In this context, the term "element" is a technical term used in XML. Elements can behave as containers to hold text, child elements, attributes, media objects or all of them.

data carousel (specified in [b-ARIB STD-B24]) and a digital video broadcasting (DVB) object carousel (specified in [b-ETSI TS 102 809]).

In the case of Ginga-HTML5, the needed IBB functionalities are enabled by remote APIs based on the representational state transfer (REST) architectural style, provided by Ginga Common Core WebServices ([b-ABNT NBR 15606-11]). Therefore, instead of JavaScript object methods or properties, Ginga-HTML5 applications access functions and data via HTTP requests and get the returned values mostly in JavaScript object notation (JSON) format. Ginga-CC WebServices provide these remote HTTP APIs to broadcaster-authorized Ginga and non-Ginga applications. In this way, any application running on devices in the home network (television (TV), SmartTV, SmartPhone etc.) may be authorized to be part of the IBB experience. Ginga-NCL [b-ABNT NBR 15606-2] applications may also access this remote API, if needed, although most IBB functionalities can also be found in its NCLua API.

If considering application conversion, the way of harmonization should convert APIs from one IBB system to another based on the equivalency described in Table 2.

Table 2 – Equivalent additional objects, properties and methods

Hybridcast		HbbTV 2.0		TOPSmedia		Ginga
Object	**Property or Method**	**Object**	**Property or Method**	**Object**	**Property or Method**	**REST API (HTTP request)**
ISDBResourceReference	original_network_id transport_stream_id service_id	Channel	onid tsid sid	Channel	number type	http(s)://<host>/dtv/current-service GET { serviceContextId, serviceName, transportStreamId, originalNetworkId, serviceId }
List of available channels can be obtained by query to a server.		ChannelList	getChannel()	ChannelList	item()	http(s)://<host>/dtv/service-list GET { { serviceContextId, serviceName, transportStreamId, originalNetworkId, serviceId }, ...}
NavigatorApplicatonManager	getOwnerApplication()	oipfApplicationManager	getOwnerApplication()	ApplicationManager	tvExt.application.appmgr	http(s)://<host>/dtv/<service-context-id>/apps GET { { appid, name, type, controlCode, state, entryPoint }, ... }
Application	replaceApplication() destroyApplication()	oipfApplication	createApplication() destroyApplication()	ApplicationManager	createApplication() destroyApplication()	http(s)://<host>/dtv/current-service/apps/<appid> POST { action: start/stop }
	visibility or display of style property of visible elements	Application	show() hide()	ApplicationManager	showApplication() hideApplication()	
KeySet	RED GREEN YELLOW BLUE NAVIGATION VCR NUMERIC value	KeySet	RED GREEN YELLOW BLUE NAVIGATION VCR NUMERIC value	ApplicationManager	keySetValue (it specifies constant values: KEY_RED, KEY_GREEN, KEY_YELLOW, KEY_BLUE, KEY_NAVIGATION, KEY_VCR, KEY_NUMERIC, KEY_OTHER)	http://<host>/dtv/current-service/ginga/keyset GET numeric_keys, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 interactive_keys, colored_keys, RED, GREEN, YELLOW, BLUE, selection_keys, ENTER, BACK, EXIT, cursor_keys, CURSOR_LEFT, CURSOR_RIGHT, CURSOR_DOWN, CURSOR_UP
	setValue()		setValue()		setKeySet()	http://<host>/dtv/current-service/ginga/keyset POST
Capabilities	hasCapability()	oipfObjectFactory	isObjectSupported()	DeviceStatus	xmlCapabilities	http(s)://<host>/dtv/platform-capabilities GET
ReceiverDevice	tuneTo()	video/broadcast object	setChannel() prevChannel() nextChannel()	HTML5 video element	src	http(s)://<host>/dtv/<service-context-id> POST
	getCurrentEventInformation()	oipfApplicationPrivateData	currentChannel	ChannelManager	getCurrentChannel()	http(s)://<host>/dtv/current-service GET

Table 2 – Equivalent additional objects, properties and methods

Hybridcast		HbbTV 2.0		TOPSmedia		Ginga
Object	**Property or Method**	**Object**	**Property or Method**	**Object**	**Property or Method**	**REST API (HTTP request)**
		ChannelConfig	currentChannel			
StreamEventTarget	addGeneralEventMessageListener() removeGeneralEventMessageListener()	video/broadcast object	addStreamEventListener() removeStreamEventListener()	Not support StreamEvent though the DSMCC (it can be substituted by W3C WebSocket or Server-Sent event on the broadband network)		http(s)://<host>/dtv/current-service/dsmcc/stream-events/<component-tag>/<carrousel-id> GET http(s)://<host>/dtv/current-service/dsmcc/stream- events/<handle> DELETE
Broadcast audio/video object	enableFullScreen()	A/V Control Object	setFullScreen()	HTML5 video/audio element	requestFullScreen()	http(s)://<host>/dtv/current-service/<comp-tag> POST { action, pos, vol } http(s)://<host>/dtv/current-service/<comp-tag> GET
		video/broadcast object	setFullScreen()			
	disableFullScreen()	A/V Control Object	setFullScreen()		exitFullScreen()	
		video/broadcast object	setFullScreen()			
	enableAudioMute()	A/V Control Object	setVolume()		muted volume	
		video/broadcast object	setVolume()			
	disableAudioMute()	A/V Control Object	setVolume()			
		video/broadcast object	setVolume()			
	isAudioMute()	video/broadcast object	getVolume()			
EITSchedule[a]	name description start_time duration service_id transportstream_id original_network_id	Programme	name description startTime duration channelID	Program	name description startTime duration channel audioInfoArray videoInfo captionInfoArray	http(s)://<host>/dtv/<service-context-id>/info/epg GET Current programme is identified in the 'present' field in the returned JSON structure
RecordedContentInformation[b]	start_time duration		selectComponent() unselectComponent() getComponents()			
ReceiverDevice[b]	getRecordedContenInformation()					
EITSearchManager	onEITScheduleUpdate() onEITSearch()	oipfSearchManager	onMetadataUpdate() onMetadataSearch()	Not support metadata search and query mechanism		http(s)://<host>/dtv/<service-context-id>/info/epg GET or
	createSearch()		createSearch()			

Table 2 – Equivalent additional objects, properties and methods

Hybridcast		HbbTV 2.0		TOPSmedia		Ginga
Object	**Property or Method**	**Object**	**Property or Method**	**Object**	**Property or Method**	**REST API (HTTP request)**
EITSearch	result setQuery() createQuery()	MetadataSearch	result setQuery() createQuery()			http(s)://<host>/dtv/all_services/info/epg GET Metadata search must be implemented by the application, based on the returned JSON structure
Query	and() not()	Query	and() not()			
SearchResults	length offset totalSize	SearchResults	length offset totalSize			
	item() getResults() abort()		item() getResults() abort()			

Notes:

[a] Since this object is a part of the results of EITSearch, no constructor is specified. However, information contained in this object is equivalent to corresponding object(s) of other systems.

[b] These objects are applicable only to recorded content.

7.2 Comparison of application life cycle aspects

7.2.1 Application life cycle

The application life cycle covers the process that takes place from the start of an IBB application to its end. To manage the application life cycle exactly, IBB systems determine each application model and the methods for starting and stopping the applications. Table 3 shows ways to launch the application of a specific type.

In addition, HbbTV 2.0 and TOPSmedia support single application execution at any given time. Thus, if an application creates a new application, the former application is terminated and the latter application launched. Even though it seems to be a transition from one application to another from the user's point of view, it is actually different from an application transition. That is, the sequence of the application terminates and launches between two different applications.

Table 3 – Launch of the application

	HbbTV 2.0	Hybridcast	TOPSmedia	Ginga
Launch the service-associated IBB application	– AUTOSTART application by broadcaster's signal (AIT) – createApplication() using 'dvb:' URL scheme listed in AIT	– AUTOSTART application by broadcaster's signal (AIT) – startAITControlledAPP()	– AUTOSTART application by broadcaster's signal (AIT) – createApplication() using 'tvapp:' URL scheme listed in AIT – User selection of installed(store) application having 'activated' attribute	– AUTOSTART application by broadcaster's signal (AIT) – Stream Event (startDocument() editing command) – User selection among the available applications (installed or signalled) via application catalogue UI – Via Ginga CC WebServices API for controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST – Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST
Launch the stand-alone IBB application	– createApplication() using HTTP/HPPTS or XML AIT – By terminal specific application (e.g., TV portal)	– startAITControlledAPP() – replaceApplication() from service-associated IBB application or stand-alone IBB application – By terminal specific application (application launcher)	– createApplication() using HTTP/HPPTS – User selection of installed(store) application having 'inactivated' attribute	– Stream Event (startDocument() editing command) – User selection among the available applications (installed or signalled) via application catalogue UI – Via Ginga CC WebServices API for Controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST – Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST

7.2.2 Application signalling

To control the life cycle of service-associated applications, the broadcaster can send the application signalling information through the current broadcast channel. HbbTV 2.0 and TOPSmedia use the application information table (AIT), with reference to [b-ETSI TS 102 809] according to each specification. Hybridcast uses the AIT specified in [b-ARIB STD-B24]. Ginga also uses the AIT, with an extended set of control codes. Table 4 shows the application control code of the systems.

According to the current AIT information sent by the broadcaster, the current application on the receiver, such as launch (autostart), keep (present) and terminate (kill), can be controlled by the broadcaster. However, in the AIT specification of HbbTV 2.0 and TOPSmedia, there is no information about the application transition that is described in the next paragraph. Thus, an application is not allowed to transit between two application types under the control of broadcaster signalling.

Table 4 – Application control code

	HbbTV 2.0	Hybridcast	TOPSmedia	Ginga
Application control code	0x01 AUTOSTART 0x02 PRESENT 0x04 KILL 0x07 DISABLED	0x01 AUTOSTART 0x02 PRESENT 0x04 KILL 0x05 PREFETCH	0x01 AUTOSTART 0x02 PRESENT 0x04 KILL	0x01 AUTOSTART 0x02 PRESENT 0x03 DESTROY 0x04 KILL 0x05 PREFETCH 0x06 REMOTE 0x07 UNBOUND 0x08 STORE 0x09 STORED_AUTOSTART 0x0A STORED_PRESENT 0x0B STORED_REMOVE

7.2.3 Application transition

Application transition occurs when the application changes its current type while keeping the current application life cycle. In other words, the change of the application type should happen in one application without any application termination and launch. Table 5 compares the application transition for each system. HbbTV 2.0 supports this with some methods for application transition between the broadcast-related and broadcast-independent applications in both directions, whereas Hybridcast and TOPSmedia only support the application transition in one direction from service-associated IBB application to the stand-alone IBB application.

Table 5 – Application transition

	HbbTV 2.0	Hybridcast	TOPSmedia	Ginga
From service associated to stand-alone	setChannel() as null for its channel argument	replaceApplication() using AIT and URL	createApplication() as an empty parameter	Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST
From stand-alone to service associated	Service selection under the conditions of clause 6.2.2.6.1 of [b-ETSI TS 102 796]	Not supported	Not supported	Via Ginga CC WebServices API for controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST

8 Harmonization of second screen handling

8.1 Communication protocols used between devices

Second screen capability can be provided in several ways. For example, relay servers for communication between devices (a TV set and second screen devices) enable a second screen IBB service without consideration of IBB system harmonization because the service can be implemented only at the application level. Therefore, all functionalities required for IBB systems involve offering communication capability between the devices and servers, which the IBB systems provide according to their specification. Further, when communicating between devices directly without a relay server, harmonization of second screen handling is required.

Direct communication between devices generally requires the use of two types of protocol between a TV set and second screen devices. Each protocol is used in each phase of device discovery and communication in order to enable services involving the second screen capability to work correctly.

Device discovery is the required action to identify the partner device for the second screen service. Various home use environments exist – there are several combinations of manufacturers, types of devices and network installations. Applications for a second screen service require seeking a partner device in various environments. Therefore, the device discovery phase is the first to establish a sufficient environment for applications. Once the application finds an appropriate partner device successfully, communications between the application or IBB system capabilities enable the launch and maintenance of the service.

These two phases are required in both cases where the protocols are proprietary or standardized. When using proprietary protocols, the IBB system software installed on both devices can be easily integrated with the proprietary services such as those provided by TV manufacturers. In addition, because a single entity such as a TV manufacturer implements the software for both devices, a lower incidence of compatibility or pairing problems can be expected. However, the UI of the software may vary depending on the implementers and thus it may be difficult to provide the same user experience from the viewpoint of IBB service providers.

When using standardized protocols, multiple implementers will develop the software for functions of device discovery and communications for applications in the same manner. Thus, there is an open environment to provide second screen services and the IBB service providers will obtain more unified UIs than in the case of proprietary protocols, provided the conformity of the software is maintained.

From an API perspective, proprietary protocols require relatively high level APIs to encapsulate the differences in the behaviours of IBB system software based on the differences in the protocols. The standardized protocols allow lower level APIs than proprietary protocols in certain cases.

8.2 Communication management

Second screen IBB services are built on top of the capability to communicate between devices. Communication management is relatively important to launch and maintain the services. The allocation of the communication manager in the software stack influences service determination flexibility.

Figure 3 a), which applies to Hybridcast and Ginga, illustrates the case where the communication manager is located in the middleware layer. In this case, an IBB application is isolated from the communication manager and hence the APIs for communication management are considered as high level APIs. Further, communication management is primarily controlled by the middleware and is transparent for the application. This location allows the communication manager to work independently of the applications. Thus, the software in the companion device can communicate with lower-layer software in the TV set, which allows the software in the companion device to have an opportunity to control the TV sets regardless of the state of execution of the application such as whether the power of the TV set is on and off, or launch control of the application in the TV set.

In Figure 3 b), which applies to HbbTV and TOPSmedia, the communication manager is located in the IBB runtime layer. This allocation is typically intended to provide APIs to control the communication manager via the application directly. Further, the application controls all behaviours of second screen services including device discovery. This structure provides flexibility for the application for services such as selection of a partner device by the application. In contrast, since the software in the companion device communicates with the manager in the IBB runtime layer, there may be difficulty in communicating with functions provided by the lower layer in the TV set.

Considering application harmonization, supplementary application programme code may be required to achieve equivalent behaviours between the cases that may or may not contain the APIs for device discovery and pairing.

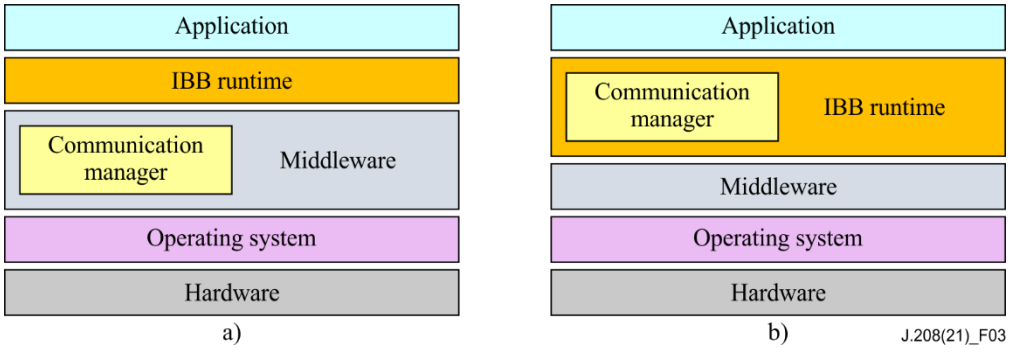

Figure 3 – Possible location of the communication manager

Bibliography

[b-ITU-T J.200] Recommendation ITU-T J.200 (2010), *Worldwide common core – Application environment for digital interactive television services*.

[b-ITU-T J.201] Recommendation ITU-T J.201 (2014), *Harmonization of declarative content format for interactive television applications*.

[b-ITU-T J.202] Recommendation ITU-T J.202 (2010), *Harmonization of procedural content formats for interactive TV applications*.

[b-ITU-T J.205] Recommendation ITU-T J.205 (2012), *Requirements for an application control framework using integrated broadcast and broadband digital television.*

[b-ARIB STD-B24] Association of Radio Industries and Businesses ARIB STD-B24 V6.4 (2017), *Data coding and transmission specification for digital broadcasting.*

[b-ARIB STD-B62] Association of Radio Industries and Businesses ARIB STD-B62 V2.2 (2019), *Multimedia coding specification for digital broadcasting (second generation).*

[b-ETSI TS 102 796] Technical Specification ETSI TS 102 796 V1.5.1 (2018), *Hybrid broadcast broadband TV*.

[b-ETSI TS 102 809] Technical Specification ETSI TS 102 809 V1.3.1 (2017), *Digital video broadcasting (DVB); Signalling and carriage of interactive applications and services in hybrid broadcast/broadband environments*.

[b-ABNT NBR 15606-2] Associação Brasileira de Normas Técnicas ABNT NBR 15606-2:2018, *Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiofusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações* [Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding].

[b-ABNT NBR 15606-11] Associação Brasileira de Normas Técnicas ABNT NBR 15606-11:2018, *Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital –Parte 11: Ginga CC WebServices – Especificação de WebServices do Ginga Common Core* [Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 11: Ginga CC WebServices – Ginga Common Core WebServices specification].

[b-IPTVFJ STD-0010] IPTV Forum Japan Standard IPTVFJ STD-0010 V2.2 (2018*), Integrated broadcast-broadband system specification*. Tokyo: IPTV Forum Japan.

[b-IPTVFJ STD-0011] IPTV Forum Japan Standard IPTVFJ STD-0011 V2.5 (2019), *HTML5 browser specification.*

[b-IPTVFJ STD-0013] IPTV Forum Japan Standard IPTVFJ STD-0013 V2.8 (2020), *Hybridcast operational guideline.*

[b-TTAK.KO-07.0111] TTA Standard TTAK.KO-07.0111/R1 (2014), *HTML5 based smart TV platform*. Seongnam-city: Telecommunications Technology Association.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021